10/867,148
10/18/04

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: 0 481 735 A2

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number: 91309496.7

(51) Int. Cl.$^5$: G06F 11/08

(22) Date of filing: 15.10.91

(30) Priority: 19.10.90 US 600512

(43) Date of publication of application:
22.04.92 Bulletin 92/17

(84) Designated Contracting States:
DE FR GB IT

(71) Applicant: ARRAY TECHNOLOGY CORPORATION
4775 Walnut Street Suite B
Boulder Colorado 80301 (US)

(72) Inventor: Brant, Bill A
4784 Dorchester Circle
Boulder Colorado 80301 (US)
Inventor: Tang, Edde Tin-Shek
4827 T-Bird Drive 5
Boulder Colorado 80301 (US)
Inventor: Hohenstein, Gerald Lee
2805 Emerson Avenue
Boulder Colorado 80303 (US)

(74) Representative: Allman, Peter John et al
MARKS & CLERK Suite 301 Sunlight House
Quay Street
Manchester M3 3JY (GB)

(54) **Address protection circuit.**

(57) An Address Protection Circuit (APC) for cross-checking the integrity of requests to read or write an addressable system memory in a fault-tolerant computer system. In the check mode, the APC checks each address and the source identification code of each memory access request from an address source. The source identification and current bus address are compared to a range of addresses stored in the APC. If the current bus address is within an "authorized" range, access to that range of locations in the memory is allowed to the address source. If a current memory access request is not authorized, the APC asserts an error signal, which may be used to transfer control to a redundant subsystem. The APC contains a content-addressable memory element that can be initialized by the subsystem processor with address ranges and type of access allowed for each source. In the setup mode, the APC is first addressed to switch the APC from its check mode to its setup mode. Thereafter, a single value in the APC tables can be changed by the processor. The APC then automatically reverts to its check mode.

EP 0 481 735 A2

Jouve, 18, rue Saint-Denis, 75001 PARIS

## BACKGROUND OF THE INVENTION

1. *Field of the Invention*

This invention relates to a fault-prevention circuit for a computer system, and more particularly to a circuit for cross-checking the integrity of Read/Write requests to an addressable system memory in a fault-tolerant computer system.

2. *Related Art*

Fault-tolerant computer systems are designed to provide "non-stop" computing despite the failure of a component, such as a circuit, power supply, or peripheral device. Such computers are often used "mission critical" applications where an interruption in computing is highly detrimental.

As principal characteristic of a truly fault-tolerant computer system is the lack of any single point of failure. That is, no failure of any single component will cause a failure of the entire system. Such fault tolerance is achieved principally by providing redundancy of function. A second important characteristic of a truly fault-tolerant computer system is the lack of any loss of data in the event of a component or subsystem failure. While "non-stop" computing is important, "error free" computing is even more so.

Redundancy can be achieved in several ways. In some instances, redundant subsystems can be operated in parallel, so that failure of one subsystem does not affect the continuing operation of the counterpart redundant subsystem. Some of such subsystems (for example, power supplies) are generally "fail safe", meaning that continued operation and data integrity are not dependent on detecting a fault condition in a failed component; when a component fails, the redundant component simply continues to provide the necessary functionality. However, with other subsystems, such as central processing units, a means must first be provided for detecting a fault. Once a fault is detected, a means must be provided for either correcting the resulting fault condition, or replacing the functionality of the failed component (e.g., with a redundant subsystem).

The principal of using redundancy to provide fault tolerance can be extended to the components of each subsystem. However, duplicating each and every component of a subsystem is expensive and adds complexity to the subsystem design. Therefore, other techniques have been developed to provide fault tolerance on a system or subsystem level without the added cost and complexity of duplicating all components. For example, solid state memory subsystems can be made fault-tolerant by adding error detection and correction circuitry implementing the well-known Hamming code. In a 32-bit wide data system, by adding only 7 additional bits to each 32-bit word permits detection of a least 2 bits in error, and correction of 1 bit in error. Thus, fault-tolerance can be achieved not by redundancy of components, but by providing redundancy of information by means of independent circuitry designed to monitor a component for failure. The independent circuitry can then either correct the error or provide some other means to accommodate the error (e.g., by providing a signal to transfer functional control to another subsystem).

The subsystems in many computer systems, such as a disk controller, contain their own microprocessor systems, typically having read-only memory (ROM), random-access memory (RAM), input/output (I/O) circuitry, and a microprocessor circuit. Fault-tolerance for the system as a whole can be achieved by providing redundant subsystems. However, provision must be made to prevent a faulty subsystem from corrupting data before a fault is detected within the subsystem. Therefore, the subsystem should be internally fault-tolerant at least to the point of not corrupting data.

Providing such internal fault-tolerance for a microprocessor subsystem presents the same issues discussed above. The components may be made redundant and operated in "lock step", so that if any one components fails, the difference between the operational states of the redundant component indicates a fault. Control may then be transferred to a redundant subsystem.

The most expensive single component of such a subsystem is the microprocessor circuit itself. A major drawback of lock-stepped microprocessors is the cost of providing a second processor and the added circuitry required to detect a difference in output of the two processors.

Therefore, it is desirable to provide some means of providing fault-detection in such a system without the added expense and complexity of redundant microprocessor circuits. The present invention provides such a means.

## SUMMARY OF THE INVENTION

The present invention comprises an Address Protection Circuit (APC) which cross-checks the integrity of requests to read or write an addressable system memory to provide fault-detection and avoid a single point of failure in a fault-tolerant computer system.

The APC provides a means for achieving subsystem-level fault-tolerance, and prevents internal subsystem data loss, based upon recognition that a fault in a processor is likely to cause a subsystem error only when the causes the processor to alter the contents of an addressable subsystem memory (such as RAM, registers, FIFO's, etc.). This concept can be generalized for subsystems in which a variety of sources can alter the contents of system memory (e.g., Direct Memory Access, or DMA, circuits).

In the preferred embodiment, the APC has two modes of operation. In the check mode, the APC is coupled to the system bus and checks each address and the source identification (SID) code of each memory access request from an address source (e.g., processor or DMA circuit). Optionally, the APC also checks the system bus Read/Write (R/W) line. The SID code and current bus address are compared to a range of addresses stored in a table in the APC. If the current bus address is within an "authorized" range, access to that range of locations in the memory is allowed to the address source. If the R/W status is also being checked, the requested access is allowed only if the current address source is addressing an authorized range of memory locations and the type of access (Read or Write) requested is authorized.

If a current memory access request is not authorized, the APC asserts an error signal. In a fault-tolerant system, the assertion of an APC error signal may be used to transfer control to a redundant subsystem.

The second mode of APC operation is a setup mode. The APC contains a content-addressable memory element that can be initialized by the subsystem processor with address ranges and type of access allowed for each address source. In the preferred embodiment, the APC must first be specifically addressed to switch the APC from its check mode to its setup mode as a safeguard against inadvertent changes to the APC authorization table. Thereafter, a single value in the APC table can be changed by the processor (whose SID code is checked before such a change is allowed). The APC then automatically reverts to its checks mode. Further changes to the APC table each require an explicit switch from the check mode to the setup mode.

The details of the preferred embodiment of the present invention are set forth in the accompanying drawings and the description below. Once the details of the invention are known, numerous additional innovations and changes will become obvious to one skilled in the art.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 is a block diagram of a microprocessor system incorporating the present invention.

FIGURE 2 is a block diagram of the preferred embodiment of the present invention.

FIGURE 3 is a diagram showing several exemplary memory locations of the content addressable memory of the preferred embodiment of the invention.

Like reference numbers and designations in the drawings refer to like elements.

## DETAILED DESCRIPTION OF THE INVENTION

Throughout this description, the preferred embodiment and examples shown should be considered as exemplars, rather than limitations on the present invention.

### OVERVIEW

FIGURE 1 shows a block diagram of a microprocessor system incorporating the present invention. A system bus 1 forms the principal communication pathway in the system, and includes data, address, and control lines. In the illustrated embodiment, addresses are 32 bits wide, corresponding to 32 address lines. The control lines include at least a Read/Write (RW) status line, an Address Strobe (AS) line, and address Source IDentification (SID) lines. The system includes at least a microprocessor 2 and system memory 3 (e.g., RAM) coupled to the system bus 1, and other components as required by a particular application. Shown in the illustrated embodiment is an I/O port 4 and disk controller 5, with attached disk 6, coupled to the system bus 1.

Also shown in FIGURE 1 is the inventive Address Protection Circuit (APC) 7, which is connected to the address lines of the system bus 1 and to an ENABLE input to the memory 3. If the APC 7 asserts a NO-ACCESS signal on the ENABLE input, the memory 3 cannot accept READ or WRITE commands, or respond to an address asserted on the system bus 1 by an address source. In addition, assertion of the NO-ACCESS signal during normal operation can be used to transfer control to a redundant subsystem.

In the illustrated embodiment, the microprocessor 2, the disk controller 5, or the I/O port 4 can be address source. Each address source is capable of asserting at least a READ or a WRITE command and an address on the system bus 1 directed to the memory 3. If access to the memory 3 is allowed by the APC 7, data can be written to or read from the memory 3 by the address source, in known fashion.

Each address source is pre-assigned a Source Identification (SID) code, which establishes a unique identity for each address source. The SID code can also be used as an access priority code to determine which address source has preferential access to the system bus 1. The SID code of each address source is asserted over the SID lines of the system bus 1. In the illustrated embodiment, the SID code is 4 bits wide, corresponding to 4 lines SID0-SID4.

In the preferred embodiment, the APC 7 has two modes of operation. In the check mode, the APC 7 is coupled to the system bus and checks each address and the SID code of each memory access request from an address source. Optionally, the APC 7 also checks the system bus RW line. The SID code and current bus address are compared to a range of addresses stored in a content-addressable memory table in the APC 7. If the current bus address is within an "authorized" range, access to that range of locations in the memory is allowed to the address source. If the RW status is also being checked, the requested access is allowed only if the current address source is addressing an authorized range of memory locations and the type of access (Read or Write) requested is authorized. If a current memory access request is not authorized, the APC 7 asserts an error signal.

The second mode of APC 7 operation is a setup mode. The content-addressable memory table in the APC 7 can be initialized by the subsystem processor with address ranges and type of access allowed for each address source. In the preferred embodiment, the APC 7 must first be specifically addressed to switch the APC 7 from its check mode to its setup mode as a safeguard against inadvertent changes to the APC 7 authorized table. Thereafter, a single value in the APC 7 table can be changed by the processor (whose SID code is checked before such a change is allowed). The APC 7 then automatically reverts to its check mode. Further changes to the APC 7table each require an explicit switch from the check mode to the setup mode.

## DETAILS OF CIRCUITRY

FIGURE 2 shows a block diagram of the preferred embodiment of the present invention. A multiplexer (MUX) 20 is coupled to two sets of inputs. In the preferred embodiment, the A-input of the MUX 20 is coupled to address lines A2-A19 of the system bus 1. The B-input of the MUX 200 is coupled to address lines A11-A23 of the system bus 1, the SID lines SID0-SID3, and the RW line. A MUXSEL signal coupled to the MUX 20 selects either the A-input or the B-input of the MUX 20 for output as an 18-bit wide CAMIN address signal.

The CAMIN output of the MUX 20 is coupled to the address inputs of a Content Addressable Ram (CAM) 21. In the illustrated embodiment, the CAM 21 comprises a 256Kx1 RAM circuit (other RAM sizes may be used with other system memory 3 sizes). The output of the CAM 21 is an ACCESS-OK signal coupled to an output Programmable Array Logic (PAL) 22. Normally, the ACCESS-OK signal is inverted and passed through the PAL 22 as the NO-ACCESS signal. As noted previously, the NO-ACCESS signal is coupled to the ENABLE input of the memory 3.

In the illustrated embodiment, the first 11 bits (A0-A11) of the address on the system bus 11 are ignored by the APC 7 during the check mode. This has the effect of treating the memory 3 as comprising "pages" of addressable locations, each page being 2048 bytes ($2^{11}$) in size. For example, an address in the range from 0 to 2047 from a single address source is treated as a single address by the CAM 21. Thus, only one memory location in the CAM 21, storing a single bit, is required to determine whether an address source is authorized to access each 2048-byte page in the memory 3. With a page size of 2048 bytes and 13 address lines applied to the B-input of the MUX 20, up to 16 MB of system memory 3 can be controlled by the APC 7. By using a larger capacity RAM circuit for the CAM 21, and/or a larger page size, a larger system memory 3 can be controlled by the APC 7. Alternatively, by using a larger capacity RAM circuit for the CAM 21, a smaller page size may be used, down to a "range" of a single byte per page.

As an added protective measure, address lines A24-A31 are input to the output PAL 22. In the illustrated embodiment, only 16MB of memory are used, requiring only 24 bits for addressing. However, the system bus 1 has 32 address lines, so the high-order 8 lines are normally unused. The PAL 22 asserts the NO-ACCESS signal if any of the address lines A24-A31 are asserted during a memory access operation. This prevents erroneous access to the memory 3, and can be used to transfer control to a redundant subsystem.

For testing and initializing purposes, it is desirable to disable the system memory 3 entirely. A third input to the output PAL 22 is a DISABLE-APC signal generated from an input PAL 23. Assertion of the DISABLE-APC signal causes the output PAL 22 to assert the NO-ACCESS signal.

Thus, in the preferred embodiments, the NO-ACCESS signal is generated under the following conditions:

$$\text{NO-ACCESS} = \overline{\text{ACCESS-OK}} \cdot \overline{\text{DISABLE-APC}} + \overline{(\text{A24-A31})}$$

An input PAL 23 generates a variety of signals within the APC 7 based upon the SID lines SID0-SID3, the RW line, the Address Strobe signal from the system bus 1 (generated by an address source when an address asserted by the address source is stable, in known fashion), a PROMSEL signal, and a CAMSEL signal. The

PROMSEL signal is generated external to the APC 7, and can be used to disable the memory 3 when a memory device (e.g., a Programmable Read-Only Memory, or PROM) other than the system memory 3 is to be addressed.

An address decoder PAL 24 accepts a single address from the microprocessor 2 (or any other processor) as a "key" address and generates the CAMSEL signal, the principal function of which is to put the APC 7 into the setup mode. The CAMSEL signal is combined with other input signals to the input PAL 23 to disable the output of the APC 7, select the A-input of the MUX 20 for input into the CAM 21, enable a tristate input buffer 25 coupled to the data input of the CAM 21 and to a data line (D0) of the system bus 1, and generate a CAMWR signal to the CAM 21 to clock data into the CAM 21. In particular, the following signals are generated by the input PAL 23 based on its input signals:

| | |
|---|---|
| MUXSEL | = $\overline{\text{CAMSEL}}$ |
| DELAY1 | = CAMSEL · AS · WR |
| DELAY2 | = DELAY1 |
| CAMWR | = CAMSEL · WR · AS · DELAY2 · (SID0-SID1 = 1110) |
| ENCAMDIN | = CAMSEL · AS · WR + DELAY1 |
| DISABLE-APC | = (CAMSEL + PROMSEL) · (SID0-SID1 = 1110) |

(it is understood in the art that the "=" sign in the above equations indicates one delay time through the PAL circuit. The DELAY1 and DELAY2 signals are internal feedback signals of the input PAL 23. For the CAMWR and DISABLE-APC signals, the address source must by the microprocessor 2, which has a SID code of "1110" in the illustrated embodiment.)

## SETUP MODE

As noted above, the contents of the memory locations of the CAM 21 are initialized during the setup mode. Each "address" (i.e., memory location) of the CAM 21 is set for one of two authorization codes: binary 0 if the current address source is not authorized for the current address range, and binary 1 if the current address source is authorized for the current address range. If the RW status is also being checked, the authorization code also depends on whether the type of access (Read or Write) requested by the address source is authorized. The CAM 21 is initialized by the microprocessor 2 (or any other processor) with address ranges and type of access allowed for each address source.

In the preferred embodiment, the APC 7 must first be addressed by a "key" address value to switch the APC 7 from its check (or "locked) mode to its setup (or "unlocked") mode. This is accomplished by the address decoder PAL 24. The address decoder PAL 24 detects a single "key" address on the address bus 1, and sets itself to hold the CAMSEL signal for the cycle of the system bus 1.

Thereafter, the address decoder PAL 24 resets itself via an internal RESET signal. The CAMSEL signal causes the next address on the address bus 1 to be applied to the CAM 21 through the A-input of the MUX 20. In addition, a data value is applied to the data input of the CAM 21 through the tristate buffer 25, and stored in the CAM 21 upon the application of the CAMWR signal from the input PAL 23.

By this sequence, each storage location in the CAM 21 can be programmed to a binary 0 or binary 1 value. However, because the CAMSEL signal is reset after initializing a CAM 21 location, the microprocessor 2 must re-supply the "key" address for each storage location to be initialized. In addition, the CAMWR signal can only be generated if the microprocessor 2 is the current address source, since the input PAL 23 checks that the SID code of the microprocessor 2 is present before generating the CAMWR signal. This security system helps insure that the APC 7 cannot be changed inadvertently, thus preserving its fault-tolerant function.

## CHECK MODE

In the check mode, the B-input is selected by the MUXSEL signal for output to the CAM 21. In effect, the address and SID code of each memory access request from an address source (e.g., microprocessor 2 or I/O port 4), and the state of the RW line, are concatenated and applied to the CAM 21 as the CAMIN address signal.

As noted above, the contents of the memory location in the CAM 21 corresponding to the CAMIN address signal is an authorization code, which is output from the CAM 21 as the ACCESS-OK signal. The authorization code is a binary 0 if the current address source is not authorized for the input address range, and a binary 1 if the current address source is authorized for the input address range. If the RW status is also being checked, as in the preferred embodiment, the requested access is allowed only if the current address source is addressing an authorized range of memory locations and the type of access (Read or Write) requested is authorized for the address source. (In some embodiments, all address sources may have both Read and Write authorization. In such a case, the WR line need not monitored.)

If the current address source is authorized to Read or Write the range of addresses in the current access request, the ACCESS-OK signal is asserted by the CAM 21. Unless blocked by the DISABLE-APC signal applied to the output PAL 22, the NO-ACCESS signal reflects the state of the ACCESS-OK signal, thus allowing the address source to access that range of locations in the memory 3.

FIGURE 3 illustrates four storage locations of the CAM 21 after initialization during the setup mode (note that the storage location addresses shown in FIGURE 3 would not be adjacent in an actual RAM circuit). CAMIN addresses 31 and 32 both represent an address range 00000000111xxxxxxxxxxx generated by address source 0111 (where each "x" indicates a "don't care" value, since the lower 11 bits of each address are not applied to the APC 7 in the check mode). In this example, a Write operation is indicated by a binary 1 on the RW status line. The contents (0) of the CAM 21 for address 31 indicates that address source 0111 is *not* authorized to Read locations in the system memory 3 in the specified range. The contents (1) of the CAM 21 for address 32 indicates that address source 0111 *is* authorized to Write locations in the system memory 3 in the same specified range.

CAMIN addresses 33 and 34 represent an address range 00000000111xxxxxxxxxxx generated by address source 1000. The contents (1) of the CAM 21 for address 32 and address 33 indicate that address source 1000 is authorized to both Read and Write locations in the system memory 3 in the specified range.

In the illustrated embodiments, additional circuitry is provided for testing purposes. As noted previously, the tristate input buffer 25 is coupled to the input of the CAM 21 and to a data line (D0) of the system bus 1. The tristate input buffer 25 is controlled by the ENCAMDIN signal generated by the input PAL 23. A tristate output buffer 26 is coupled to the output of the CAM 21 and to a data line (D0) of the system bus 1. The tristate output buffer 26 is controlled by an ENCAMDOUT signal generated by the input PAL 23:

$$\text{ENCAMDOUT} = \text{CAMSEL} \cdot \overline{\text{WR}}$$

Normally, both the input buffer 25 and the output buffer 26 are disabled (i.e., placed in the high-impedance state). In the setup mode of the APC 7, or in a test mode, ENCAMDIN is asserted by the input PAL 23, allowing data to be loaded into the CAM 21 through the input buffer 25. In the test mode, if ENCAMDOUT is asserted, the output of the CAM 21 can be directly fed back to the system bus 1 through the output buffer 26. This arrangement permits data patterns to be loaded into the CAM 21 and directly read out for comparison by the microprocessor 2. Any discrepancies indicate a likely fault in the APC 7 or the system bus 1.

A number of embodiments of the present invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. For example, the CAM 21 may be programmed over a separate bus from the system bus 1. In appropriate applications, the CAM 21 may be a pre-programmed ROM or PROM, and thus much or the other circuitry shown in FIGURE 2 would be unnecessary. Further, while a content-addressable memory is the preferred means of storing the authorization table of the APC 7, other "look-up" table circuit structures may be used, or alternatively, algorithmic means (e.g., hashing algorithms) may be used, to check address source addresses for authorization. Accordingly, it is to be understood that the invention is not to be limited by the specific illustrated embodiment, but only by the scope of the appended claims.

## Claims

1. An address protection circuit for detecting erroneous memory address requests from at least one address source to an addressable memory, the address protection circuit including an address validation means, coupled to at least one address source and to the addressable memory, for generating and transmitting an access approval signal to the addressable memory upon receipt from any address source of a valid memory request.

2. The address protection circuit of claim 1, wherein the address validation means includes an address comparison means for comparing a memory address request from any address source to a set of predefined authorization codes corresponding to address ranges in the addressable memory, and for generating and transmitting the access approval to the addressable memory upon receipt from any address source of a memory address request corresponding to a least one such authorization code.

3. The address protection circuit of claim 1, wherein the address validation means includes a data storage means for storing at least one authorization code for a corresponding address range in the addressable memory, and for generating and transmitting the access approval signal to the addressable memory upon receipt from any address source of a memory address request corresponding to one such authorization code.

4. The address protection circuit of claim 3, wherein the data storage means is content addressable.

5. The address protection circuit of claim 2, further including programming means, coupled to the address comparison means, for selectively defining each authorization code.

6. The address protection circuit of claim 5, wherein the programming means includes security means for limiting access to the programming means to a preselected address source.

7. The address protection circuit of claim 3, further including programming means, coupled to the data storage means, for selectively defining each authorization code.

8. The address protection circuit of claim 7, wherein the programming means includes security means for limiting access to the programming means to a preselected address source.

9. The address protection of claim 2, further including disabling means, coupled to the address comparison means, for preventing the transmittal of the access approval signal to the addressable memory.

10. The address protection circuit of claim 3, further including disabling means, coupled to the data storage means, for preventing the transmittal of the access approval signal to the addressable memory.

11. The address protection circuit of claim 2, further including testing means, coupled to the address comparison means and to an address source, for transmitting the generated access approval signal to the address source upon receipt by the address comparison means of a memory address request from any address source.

12. The address protection circuit of claim 3, further including testing means, coupled to the data storage means and to an address source, for transmitting the generated access approval signal to the address source upon receipt by the address comparison means of a memory address request from any address source.

13. The address protection circuit of claims 1, 2, 3, wherein each memory address request from an address source includes a set of addressable memory address signals and a source identification code identifying the address source.

14. The address protection circuit of claim 13, wherein each memory address request further includes a signal indicating an input/output operation.

15. An address protection circuit for detecting erroneous memory address request from at least one address source to an addressable memory, the address protection circuit including content addressable storage means, coupled to at least one address source and to the addressable memory,
a. for storing at least one authorization code for a corresponding address range in the addressable memory, and
b. for generating and transmitting an access approval signal to the addressable memory upon receipt from any address source of a memory address request corresponding to one such authorization code;
wherein each memory address request from an address source includes a set of addressable memory address signals and a source identification code identifying the address source.

16. The address protection circuit of claim 15, further including programming means, coupled to the content addressable storage means and to an address source, for selectively defining each authorization code.

17. The address protection circuit of claim 16, wherein the programming means includes security means for limiting access to the programming means to a preselected address source.

18. The address protection circuit of claim 15, wherein each memory address request further includes a signal indicating an input/output operation.

19. The address protection circuit of claim 15, further including disabling means, coupled to the content addressable storage means, for preventing the transmittal of the access approval signal to the addressable

memory.

**20.** The address protection of claim 15, further including testing means, coupled to the content addressable storage means and to an address source, for transmitting the generated access approval signal to the address source upon receipt by the address comparison means of a memory address request from any address source.

**21.** An address protection circuit for detecting erroneous memory address requests from at least one address source to an addressable memory, the address protection circuit including:

a. content addressable storage means, coupled to at least one address source and to the addressable memory,

(1) for storing at least one authorization code for a corresponding address range in the addressable memory, and

(2) for generating and transmitting an access approval signal to the addressable memory upon receipt from any address source of a memory address request corresponding to one such authorization code;

b. programming means, coupled to the content addressable storage means and to an address source, for selectively defining each authorization code;

c. security means, coupled to the programming means, for limiting access to the programming means to a preselected address source; and

d. disabling means, coupled to the content addressable storage means, for preventing the transmittal of the access approval signal to the addressable memory;

wherein each memory address request from an address source includes a set of addressable memory signals, a source identification code identifying the address source, and a signal indicating an input/output operation.

**22.** A method for detecting erroneous memory address requests from at least one address source to an addressable memory, comprising the steps of:

a. receiving memory address requests from at least one address source;

b. generating and transmitting an access approval signal to the addressable memory upon receipt from any address source of a valid memory address request.

**23.** A method for detecting erroneous memory address requests from at least one address source to an addressable memory, comprising the steps of:

a. receiving memory address requests from at least one address source;

b. comparing the received address request to a set of predefined authorization codes corresponding to address ranges in the addressable memory,

c. generating and transmitting the access approval signal to the addressable memory upon receipt of a memory address request corresponding to the least one such authorization code.

**24.** The method for detecting erroneous memory address requests of claim 23, further including the step of selectively defining each authorization code.

**25.** The method for detecting erroneous memory address requests of claim 24, further including the step of limiting the definition of each authorization code to a preselected address source.

**26.** The method for detecting erroneous memory address requests of claim 23, further including the step of selectively preventing the transmittal of the access approval signal to the addressable memory.

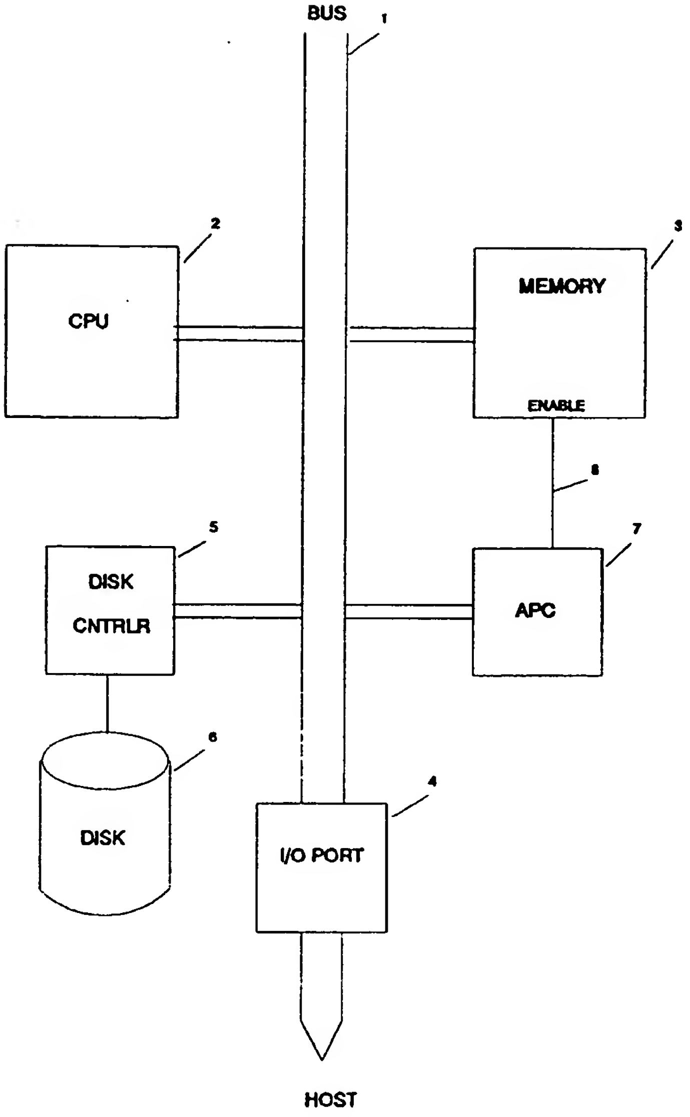

BUS
1
2
CPU
3
MEMORY
ENABLE
8
5
DISK
CNTRLR
7
APC
6
DISK
4
I/O PORT
HOST

FIGURE 1

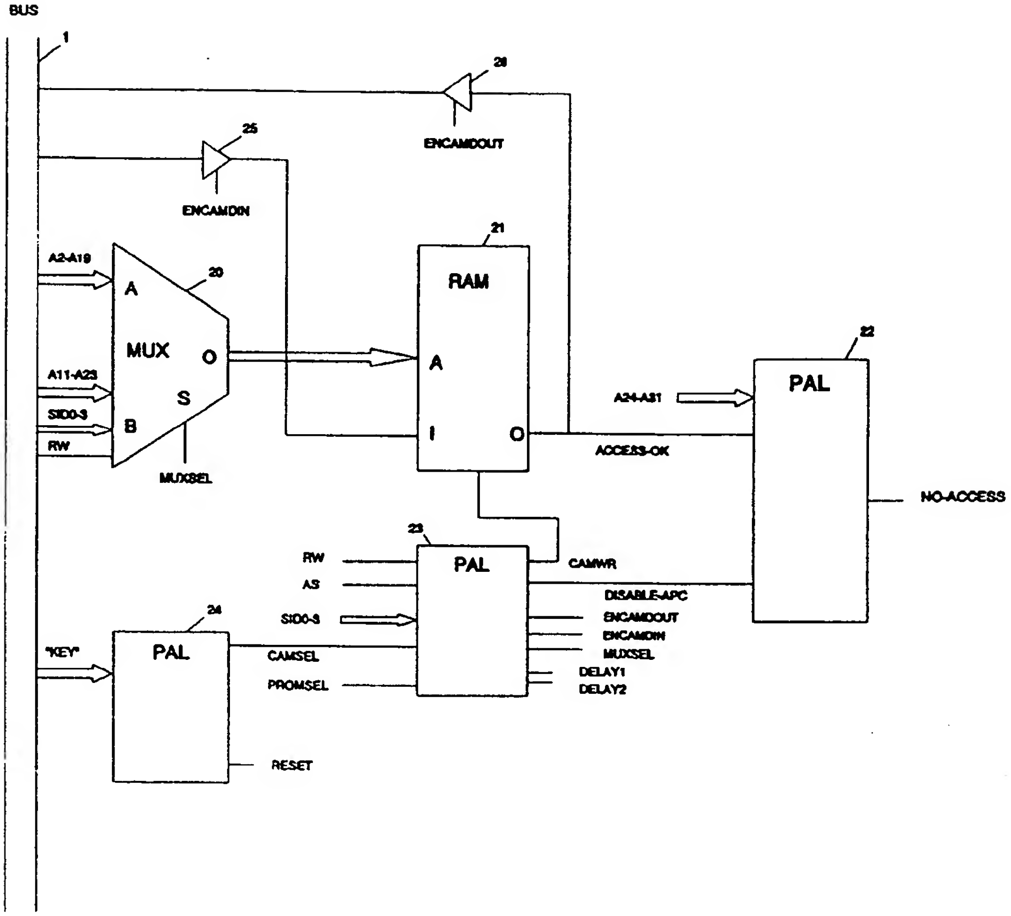
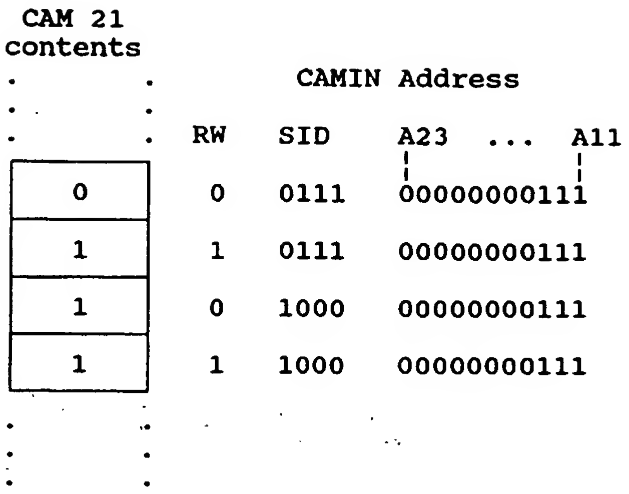

BUS
1
28
ENCAMDOUT
25
ENCAMDIN
21
RAM
A2-A19
20
A
MUX
O
A11-A23
S
SIO0-3
B
RW
MUXSEL
A
I
O
22
PAL
A24-A31
ACCESS-OK
NO-ACCESS
23
RW
AS
PAL
CAMWR
DISABLE-APC
SIO0-3
ENCAMDOUT
ENCAMDIN
MUXSEL
DELAY1
DELAY2
24
"KEY"
PAL
CAMSEL
PROMSEL
RESET

FIGURE 2



| CAM 21 contents | CAMIN Address |  |  |
|---|---|---|---|
|  | RW | SID | A23 ... A11 |
| 0 | 0 | 0111 | 00000000111 |
| 1 | 1 | 0111 | 00000000111 |
| 1 | 0 | 1000 | 00000000111 |
| 1 | 1 | 1000 | 00000000111 |

FIG. 3